EARTHQUAKE ENGINEERING AND STRUCTURAL DYNAMICS
Earthquake Engng. Struct. Dyn. **28**, 995–1018 (1999)

EXPERIMENTAL STUDY OF IDENTIFICATION AND CONTROL OF STRUCTURES USING NEURAL NETWORK PART 1: IDENTIFICATION

KHALDOON BANI–HANI[†], JAMSHID GHABOUSSI[‡,*] AND STEPHEN P. SCHNEIDER[¶]
Department of Civil Engineering, University of Illinois at Urbana Champaign, Urbana, IL 61801, U.S.A.

SUMMARY

Experimental verifications of a recently developed structural control method using neural network has been carried out on the earthquake simulator at the University of Illinois at Urbana—Champaign. The test specimen was a 1/4 scale model of a three-storey steel frame. The control system consisted of a tendon/pulley system controlled by a single hydraulic actuator. The model structure had a total mass of 2994 kg (6600 lb), distributed evenly among the three floors, and a total frame height of 254 cm (100 inches). The structure had three distinct lightly damped fundamental modes of vibration plus two higher modes representing the structure–control interaction and the actuator dynamics. The system identification and parameter estimation have been conducted in two experimental methods: first, the system has been identified in the time domain and the estimated parameters were used in the frequency domain methods and secondly, the system was modelled and identified using multiple emulator neural networks with different prediction capabilities. These emulators were employed in the control design. This paper describes the test set-up, the experimental validation of the identified model in the time and frequency domains, and experimentally demonstrates the performance of the multiple emulator neural networks. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: structures; dynamics; neural networks; identification; earthquake engineering

INTRODUCTION

The effectiveness, robustness and stability of the controllers depends on the accuracy of the dynamic system models. In conventional control methods and control design is highly dependent on the parametric construction of the dynamic system models. Therefore, in conventional control methods, such as the optimal control method the designer must identify the controlled system accurately. In actuality control systems are inherently non-linear, and therefore it is difficult, if not impossible, to build the real parametric non-linear system models. Consequently, in conventional

*Correspondence to: Jamshid Ghaboussi, Department of Civil Engineering, University of Illinois at Urbana Champaign, 3118 Newmark Civil Engineering Laboratory, MC-250, 205 North Mathews Ave., Urbana, Illinois 61801-2397, USA.

[†]Assistant Professor, Jordan University of Science and Technology, Jordan
[‡]Professor of Civil Engineering
[¶]Associate Professor of Civil Engineering

CCC 0098–8847/99/090995–24$17·50

Received 17 September 1998
Accepted 25 March 1999

control methods the non-linear systems are approximated by linear dynamic models. These methods have produced successful results in structural control to date.[1–4] This paper presents a new neural network based method to enhance the system identification techniques. This methodology enables the controller to include non-parametric neural network identifiers to represent the non-linear behaviour. In the last few years neural networks have shown promising capabilities in the application of system identification and robust control of linear and non-linear systems.[5,6] This paper and a companion paper describe a comprehensive experimental study that has been conducted in an effort to verify a recently proposed neural network based approach in structural control. This study includes the first experimental research in the use of neural networks in structural control.

In this study two different control strategies have been considered, verified experimentally and compared. First, a neural network based structural control approach has been used to develop the first set of two neurocontrollers[7–9] with different architecture, feedback and sampling period. Second, a linear quadratic optimal controller with acceleration feedback has been designed and tested.

This paper presents a full description of the test set-up, the test specimen and control system, and presents the details of the system identification and parameter estimation experiments, where the dynamic system model has been constructed in a parametric fashion for the optimal control design and in parametric/non-parametric fashion for the neurocontrollers design. In the parametric models, the system identification has been conducted in the time domain using experimentally generated data and next the estimated parameters were used to enhance the analytical model in the frequency domain. In the non-parametric identification, 'black-box' model, neural networks were used to emulate the system behaviour and to identify transfer functions from the experiments. To accomplish this goal, multiple emulator neural networks have been trained and designed. These emulators have different prediction capabilities with different time delays and different sampling periods.

The experiments were organized in three successive phases. The first phase was a series of experiments for system identification and parameter estimation. A mathematical parametric state space model was developed. The validation results of the simulation and the experiments are presented and compared. The second phase contained four shaking table tests that were used for the training of the multiple emulator neural networks which identify the system transfer functions and predict the structure response. The effectiveness of these emulator neural networks were evaluated and presented experimentally. The last phase was the development of the controllers: optimal controller and two neurocontrollers. The first and second phases of the study are presented in this paper while the last phase has been discussed in a companion paper.[9]

EXPERIMENTAL SET-UP

The earthquake simulator

The experiments were conducted on the earthquake simulator at the University of Illinois at Urbana–Champaign as shown in Figure 1. The earthquake simulator is anchored to a strong structural test floor, and provides dynamic base excitation in one horizontal direction. The earthquake simulator has a platform area of $3{\cdot}66 \times 3{\cdot}66\ m^2$ ($12 \times 12\ ft^2$) in plan and comprises a 0·95 cm (3/8″) plate welded to 12·7 cm (5″) I–beams with system natural frequency of 0·4 Hz. The

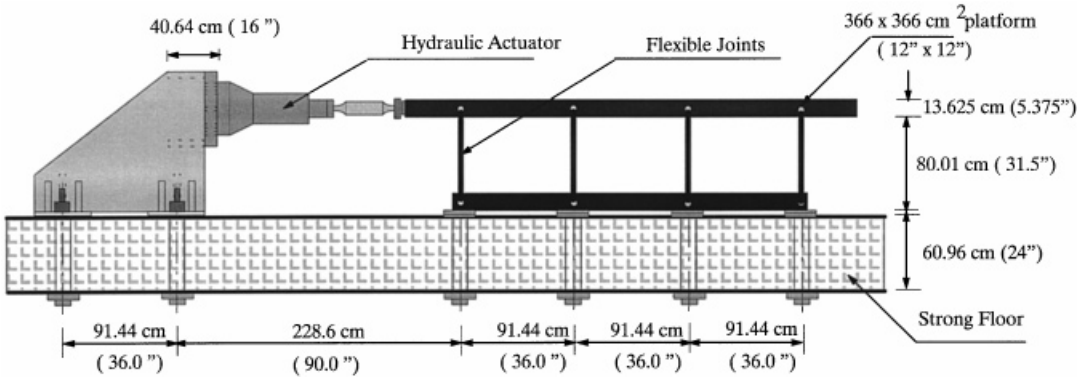


Figure 1. The earthquake simulator at the University of Illinois at Urbana—Champaign

shake table is connected to a 333·6 KN (75 kip) hydraulic actuator capacity with a 10·16 cm (4 in) piston stroke length. The platform has been designed to support test specimens up to 4·536 ton (10 kip). The table actuator is controlled with MTS 469 control system with MTS model 252 two-stage servovalve with capacity of 30 to 400 gpm. A 120 hp hydraulic power supply with 70 gpm hydraulic pump and 20 gpm hydraulic pump that give a total continuous hydraulic flow of 90 gpm. A hydraulic service manifold and accumulates are utilized for hydraulic pressure storage and to furnish short-term, high-flow demands to the servovalve. The feedback control of the shake table is accomplished with the MTS three-variable control (TVC) servo circuitry, the essence of this technique is the simultaneous control of displacement, velocity and acceleration. The maximum table velocity is up to 38·1 cm/sec (15 in/sec) and the peak acceleration is approximately 7·5 g. The earthquake simulator specifications are discussed by Sozen et al.[10] and the performance limits are reported to be between 1–100 Hz.

The experimental specimen

The test specimen is a three-storey steel frame as shown in Figure 2. The specimen was designed to approximate similar structural models tested in different studies.[11] The structural system was an approximate 1/4 scale model of a prototype structure which has been introduced by Clough and Tang[12] for seismic testing. This model has become the standard for structural control problems,[13] and has been used in many structural control tests.[1,4] The model has a total mass of 2994 kg (6600 lb), distributed evenly among the three floors. The structure has three distinct lightly damped fundamental modes of vibration plus two higher modes that represent the building–control system interaction and the actuator dynamics. The steel frame was constructed using structural tubing of 5·08 cm (2″) and 0·3175 cm (1/8″) thickness. The model was mounted on the earthquake simulator such that the two moment-resisting frames were parallel to the axis of the motion of the table. Braces were used along the orthogonal axis of the model to minimize out-of-plane deformations. Structural tubing of 5·08 cm (2″) and 0·635 cm (1/4″) thickness were used for braces. The steel frame model was bolted to the shake table platform using (16) 1/2″ A325 bolts using four base plates welded to the four columns of the frame.

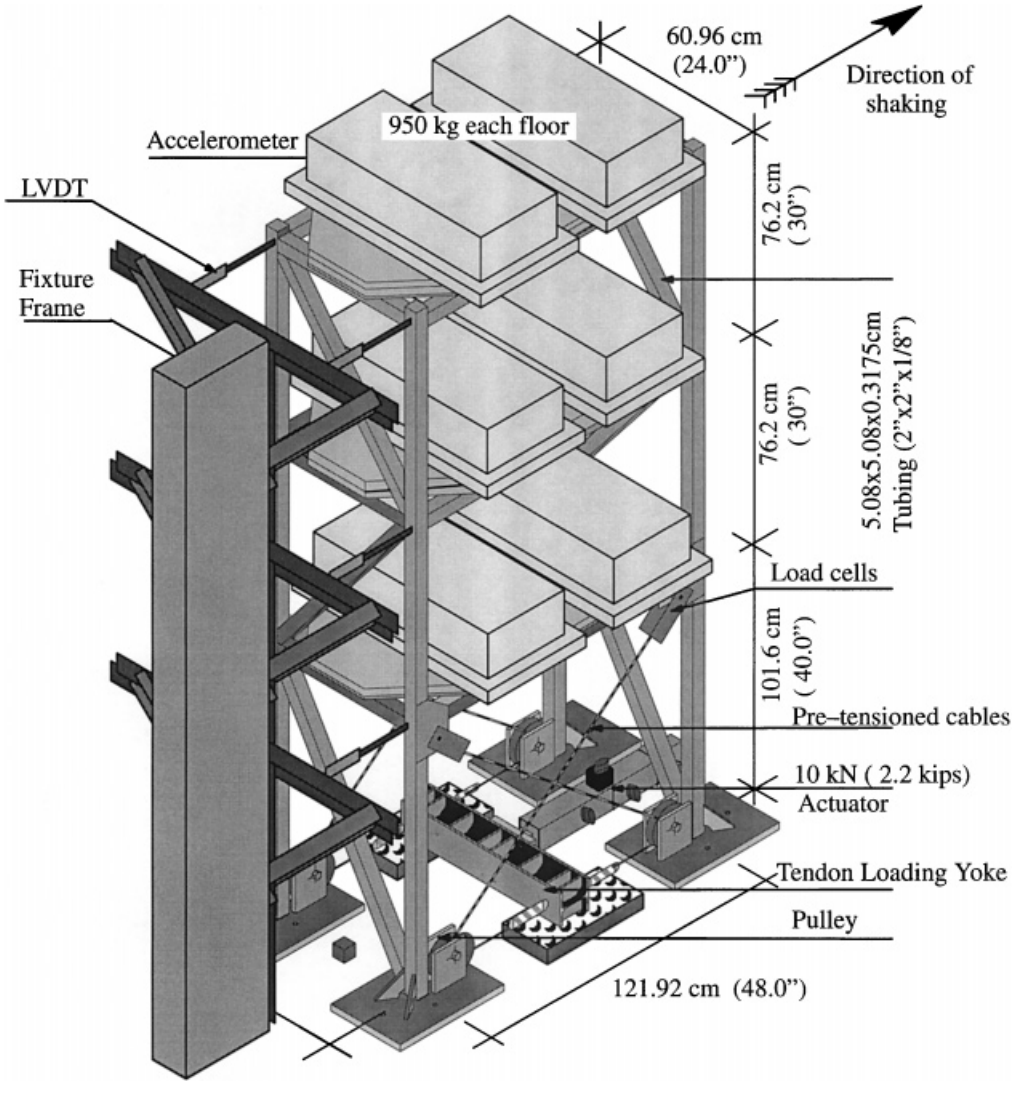


Figure 2. The test specimen used in this study. Three storey steel frame with active tendons

The control system consists of four 0·635 cm (1/4″) Φ 6X37IWRC wire rope tendons. Uniaxial tension tests were conducted on the wire ropes and the average yielding capacity was found to be approximately 20 000 N (4500 lb) and the average stiffness of these wires have been estimated to be about 9631 N/cm (5500 lb/in). The four tendons were pre-tensioned to 8896 N (2000 lb) force to ensure the continuity of tension force in the tendons during the tests. Four pulleys of 10·478 cm (4 1/8″) diameter each and service strength of 11 200 N (2500 lb) were bolted at the base plates to direct the control force to the first floor. The tendons are connected to the first floor through a set of steel brackets. A W5X16 load yoke was used to transfer the actuator force to the four tendons

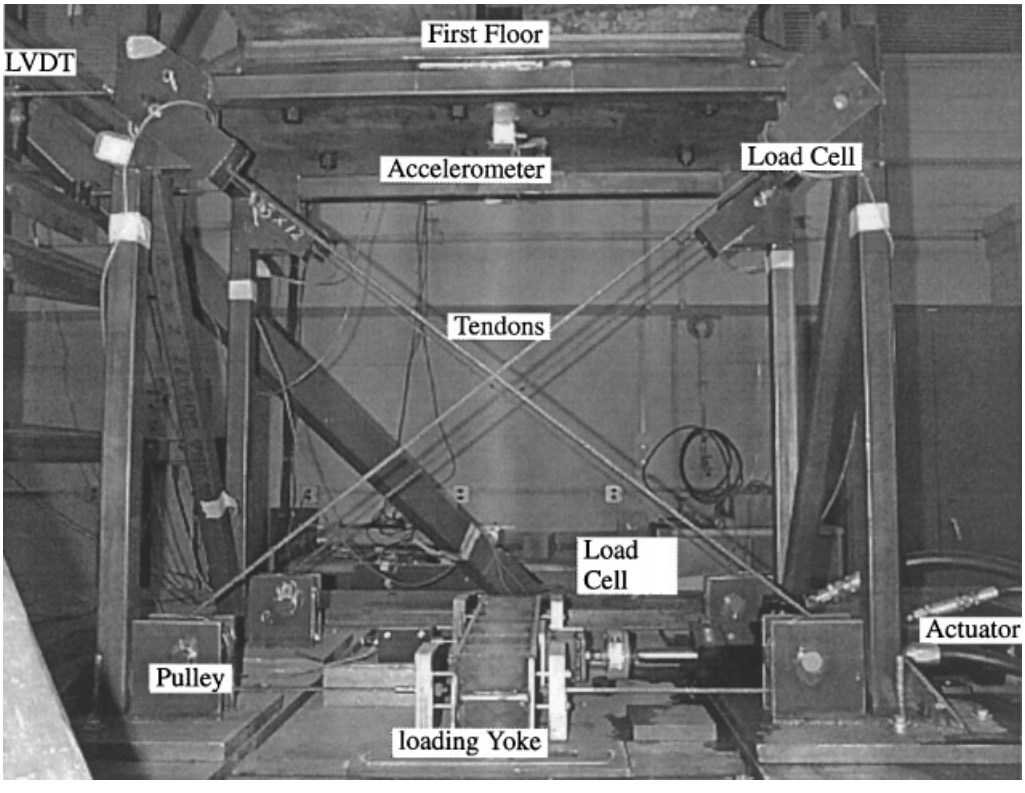


Figure 3. The control system elements

through a special connection. The tendons transferred the actuator forces to the first floor from the loading yoke through the pulley system. A computer controlled hydraulic actuator of 9785 N (2200 lb) capacity is connected to the loading yoke. Consequently, the hydraulic actuator at the base coupled with the tendon–pulley system controlled the motion of the first floor of the structure. A detailed photo illustrating these elements is shown in Figure 3.

Instrumentations

Strategically placed measurement transducers were mounted on the test specimen to provide an accurate representation of the structure motion. Six Linear Variable Differential Transforms (LVDTs) were attached to the structure and connected to a rigid reference frame mounted on the earthquake simulator platform to measure the relative displacements of each floor. Each floor had two LVDTs to measure the average lateral displacement and any possible torsional deformation. Three accelerometers were installed, one for each floor, midway between the two frames, to measure the absolute accelerations of the three floors. One accelerometer was mounted on the earthquake simulator to measure the actual ground acceleration. Four load cells connected to the end of each tendon measured the change in the tendon force where the sum of these changes represented the applied control force. A load cell was connected between the loading yoke and the

control actuator to measure the control force. The internal LVDT in the hydraulic actuator was used to measure piston movement and to stabilize the actuator control closed-loop.

The control actuator has been regulated through a Power PC computer and an INSTRON Plus 8500 digital unit through the NB-MIO-16X board. The NB-MIO-16X is a high-performance multifunction analog, digital and timing input/output (I/O) board for Macintosh NuBus computers. The NB-MIO-16X contains a 16-bit Analog-to-Digital Converter (ADC) with up to 16 analog inputs, and 12-bit Digital-to-Analog Converter (DACs). The actuator control loop was stabilized by using the actuator displacement feedback through the INSTRON controller.

PARAMETRIC SYSTEM IDENTIFICATION

Generally, the mathematical model of the structural system can be obtained by two different methods. The first method is to generate a model based on the physical properties of the system. These analytical finite element models can be developed using mass and stiffness characteristics of the structure. However, this leads to an approximate, incomplete system model which may be adequate for analysis and design purpose but not for structural control. A more accurate and reliable method, known as the system identification technique, is to develop the model based on experimental data that possess the real system behaviour. In this study parametric modelling of the test system has been achieved by two experimentally-based methods, the time domain[14] and the frequency domain[15,16] methods. Both methods were used in the system identification and parametric estimations. First, the modelling was performed in the time domain. Next the estimated parameters and the system model determined in the first step were employed to produce a more accurate system model using the frequency domain methods.

In the system identification a number of necessary steps must be performed to obtain the best fit models that describe the dynamic system with sufficient accuracy. First, the designer must choose the class of the models to be identified: this covers the order of the model, the application requirements, the required accuracy, whether the behaviour is linear or non-linear, and whether the model is parametric or non-parametric. The choices are governed by the physical behaviour of the system, the application and the insight and skill of the designer. The second step represents the estimation of the parameters of the selected model: there are different approaches to accomplish this goal, most of these approaches seek to minimize the difference between the measured data and the modelled data, while minimizing the external noise effects. The third step is the testing of the identified model; it is checked for the residuals produced from the measured data used in the previous step which can be due to unmodelled dynamics, noise or non-linear behaviour. The final step is the validation and generalization of the model for new experimental data.

One important issue in the system identification is the generation of the experimental data used in the parameter estimation approach. This task can be achieved by exciting the system with signals that can produce the system output in the desired range of frequencies. In designing the excitation signal and recording the output signals different concepts and issues have to be controlled and observed: *aliasing*, *window functions*, *digital filters*, *sampling time*, *and filter leakage*. These concepts are discussed in many vibration testing books, such as McConnell.[17]

In this study the experimental data generated for the system identification was collected in two separate tests conducted at 200 kHz sampling rate. In the first test, the system was only excited with white noise ground motion of (0–60 Hz). The identified transfer functions were used to build

a linear subsystem with one input (ground excitation) and many outputs (system response). In the second test, the system was excited by band-limited white noise (0–60 Hz) control signal to determine the relationship between the control signal and the system response. This resulted in a second linear subsystem with the different input (the control signal) and the system response output. The previous concepts mentioned in developing the excitation signal were observed and satisfied. Moreover the developed experimental data were digitally processed and filtered using a Butterworth digital lowpass *anti-aliasing* filter with 60 Hz cut-off frequency.

Time domain techniques

Practical limitations on data acquisition allow sampling the response of a test specimen at discrete time intervals only. Thus the output response $y(t)$ is observed at a discrete instants $t_k = kT_s$, $k = 1, 2, \ldots, n$, and T_s is the sampling period. Moreover, the input signal is often kept constant between the sampling period in the computer implementations. Consequently, in the discrete-time domain, the input–output model can be described in the form of the general Non-linear AutoRegressive Moving Average (NARMA) model, where the new system output can be predicted using the past input and output samples of the system. In general, the input–output system can be described as follows:

$$y(k) = f(y(k-1),\ y(k-2),\ \ldots,\ y(k-n),\ u(k),\ u(k-1),\ \ldots,\ u(k-m)) \tag{1}$$

where $y(k)$ and $u(k)$ represent the input–output pair of the system at discrete time step k, and the positive integers n and m are the order of the system. If the system is assumed to be linear, the function f is linear. Thus, the system can be described as follows:

$$\boldsymbol{y}(k) = \sum_{i=1}^{n} a_i \boldsymbol{y}(k-i) + \sum_{j=0}^{m} b_j \boldsymbol{u}(k-i) \tag{2}$$

Equation (2), can be rewritten in a compact vector form as follows:

$$y(k) = \boldsymbol{\beta}(k)\boldsymbol{\theta} \tag{3}$$

where

$$\boldsymbol{\theta} = [a_1, a_2, a_3, \ldots, a_n, b_0, b_1, \ldots, b_m]^{\mathrm{T}} \tag{4}$$

$$\boldsymbol{\beta}(k) = [y(k-1), y(k-2), \ldots, y(k-n), u(k), u(k-1), \ldots, u(k-m)] \tag{5}$$

So the best least-squares fit of the parameter vector, $\hat{\boldsymbol{\theta}}$, can be estimated over the experimental input–output data. Then, if we define

$$\boldsymbol{Y}_N = [y(1), y(2), \ldots, y(N)]^{\mathrm{T}} \text{ and } \boldsymbol{\phi}_N = [\boldsymbol{\beta}(1), \boldsymbol{\beta}(2), \ldots, \boldsymbol{\beta}(N)]^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{Y}_N = \boldsymbol{\phi}_N \boldsymbol{\theta} \tag{7}$$

where N is the number of sampling times, then the estimated response based on the least-squares fit $\hat{\boldsymbol{\theta}}$ can be defined as

$$\hat{\boldsymbol{Y}}_N = \boldsymbol{\phi}_N \hat{\boldsymbol{\theta}} \tag{8}$$

The total error through the input–output data for the estimation is defined as

$$\boldsymbol{E}_N = [\boldsymbol{Y}_N - \boldsymbol{\phi}_N \hat{\boldsymbol{\theta}}]^{\mathrm{T}} [\boldsymbol{Y}_N - \boldsymbol{\phi}_N \hat{\boldsymbol{\theta}}] \tag{9}$$

By taking the gradient of the error function with respect to $\hat{\boldsymbol{\theta}}$ and setting it equal to zero

$$\nabla_{\theta} \boldsymbol{E}_N = \mathbf{0} = -2\boldsymbol{\phi}_N^{\mathrm{T}}[\boldsymbol{Y}_N - \boldsymbol{\phi}_N \hat{\boldsymbol{\theta}}] \tag{10}$$

The least-squares fits $\hat{\boldsymbol{\theta}}$ becomes

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\phi}_N^{\mathrm{T}} \boldsymbol{\phi}_N)^{-1} \boldsymbol{\phi}_N^{\mathrm{T}} \boldsymbol{Y} \tag{11}$$

when the inverse of the coefficient matrix exists (otherwise pseudo-inverse is used).

Equation (11) can be solved in a recursive routine for simple computations and real time identification. This can be done by some matrix manipulation of equation (11) when the estimated parameter $\hat{\boldsymbol{\theta}}_{N+1}$ at batch set $N+1$ can be estimated from the previous parameter $\hat{\boldsymbol{\theta}}_N$ at batch N. This is described in equation (12).

$$\hat{\boldsymbol{\theta}}_{N+1} = \hat{\boldsymbol{\theta}}_N + \frac{(\boldsymbol{\phi}_N^{\mathrm{T}} \boldsymbol{\phi}_N)^{-1} \beta^{\mathrm{T}}}{1 + \beta(\boldsymbol{\phi}_N^{\mathrm{T}} \boldsymbol{\phi}_N)^{-1} \beta^{\mathrm{T}}} [\boldsymbol{y}(N+1) - \beta \hat{\boldsymbol{\theta}}_N] \tag{12}$$

A MATLAB[18] code was used to fit the strategic relationships of the input–output system according to equation (12). These relationships where chosen to describe the system motion sufficiently. Sufficient input–output data, ten tests of 60 s duration each, were collected experimentally and used for system modelling. The estimated parameter were used to extract the fundamental frequencies and the corresponding damping, which have been used as initial estimation in the next step, the frequency domain analysis.

Frequency domain techniques

In the frequency domain identification method, the experimental transfer functions of the input–output relationships were estimated, empirically, from ten experiments on the shake table for the two different cases; the ground excitation and the actuator command. The transfer functions were estimated by computing the fast Fourier transform of the inputs and the outputs. Then, the power spectral density of the input signal $\mathrm{PSD}_{uu}(\mathrm{j}w)$, and the cross spectral density of the input/output signal $\mathrm{CSD}_{uy}(\mathrm{j}w)$, were used to obtain the empirical transfer functions defined as follows.[14]

$$G_{yu}(\mathrm{j}w) = \frac{\mathrm{CSD}_{uy}(\mathrm{j}w)}{\mathrm{PSD}_{uu}(\mathrm{j}w)} \tag{13}$$

The transfer function of each input–output pair is estimated in each experiment, and the final estimation is obtained as the average over the ten experiments. A least-squares routine is used to determine the best fit $\hat{G}_{yu}(\mathrm{j}w)$ of these transfer functions making use of the estimated parameters in the previous step in the time domain identifications.[15] The error function is defined as follows:

$$E = \sum_{k=1}^{n} |\hat{G}_{yu}(\mathrm{j}w) - G_{yu}(\mathrm{j}w)|^2. \tag{14}$$

The transfer function can be rewritten furthermore in the form of its numerator and denominator as follows:

$$G_{yu}(\mathrm{j}w) = \frac{\mathrm{NUM}(\mathrm{j}w)}{\mathrm{DEN}(\mathrm{j}w)} = \frac{\sum_{i=1}^{n} \alpha_i (\mathrm{j}w)^i}{\sum_{i=1}^{d} \beta_i (\mathrm{j}w)^i} \tag{15}$$

A least-square fit is used to find the transfer function parameters a_i and β_i. A MATLAB[18] code was used to fit the specified transfer functions making use of the estimated parameters from the time domain analysis.

State space representation

Once all the desired transfer functions were identified, the zeros, poles and gains of each transfer function were computed. The transfer functions are assumed to have common poles. However, the least-squares fit does not necessary produce the same poles for all the transfer functions, so an adjustment of the locations of the poles and zeros was performed to represent the system more precisely. These transfer functions, as mentioned before, described two linear subsystems of different inputs and common outputs. Combining the two subsystems has been done by transforming each one of them into state space form, then connecting both of the state spaces in *parallel* form. This was possible since the system was linear and superposition still applies. However, the *parallel form* does not produce the minimum realization of the state space system. Therefore, further effort was needed to produce the minimum realization of the state space system by performing a balanced realization and model reduction on the state space. This has been accomplished using the tools available in the MATLAB[18] software.

The system identification process and the state space realization resulted in high-fidelity model of 25 poles corresponding to system model up to 100 Hz. However, ten of these poles represent the lightly damped fundamental poles of the system: six poles represent the structural fundamental poles, $-0{\cdot}0321 \pm 23$j, $-1{\cdot}18 \pm 71{\cdot}5$j, $-0{\cdot}0754 \pm 129$j (in rad/sec), and four poles that represent the control system and the actuator dynamics, $-1{\cdot}48 \pm 288$j, and $-2{\cdot}17 \pm 314$j (in rad/sec). These poles correspond to natural frequencies of 3·66, 11·38, 20·53, 45·84 and 49·98 Hz, and corresponding dampings are 1·39, 1·65, 0·586, 0·513 and 0·691 per cent. In the sense of model accuracy, the analytical model was assumed to represent the actual model accurately for the frequencies below 50 Hz, which is in the range of interest for this system.

Mathematical model validations

The validity of the identified model will be illustrated by presenting sample comparisons of experimental and analytical results. Figure 4 shows a comparison between the experimental and analytical transfer functions from the ground excitation $\ddot{\boldsymbol{x}}_g$ to the first and the third floor absolute accelerations. Figure 5 shows a sample comparison between the experimental and analytical transfer functions from the actuator command u to the first and the third floor absolute accelerations. It can be seen that the analytical model successfully represented the needed transfer functions. As for the distinct lightly damped fundamental modes, Figure 5 clearly shows the five peaks representing the ten poles which were mentioned earlier. Figure 6 shows a comparison of the experimental and model simulation results of the third floor absolute acceleration for the three loading cases: (a) 0–50 Hz band-limited white noise actuator command, (b) 0–50 Hz band-limited white noise ground acceleration, and (c) ground acceleration of 50% of the N21°E component of the 1952 Taft earthquake record, measured at the Lincoln School Tunnel. In the model validation tests, it is worth noting that the validation experiments were generated from different excitations than the ones used in the system identification. This was to demonstrate that the identified model was independent of the input excitation.

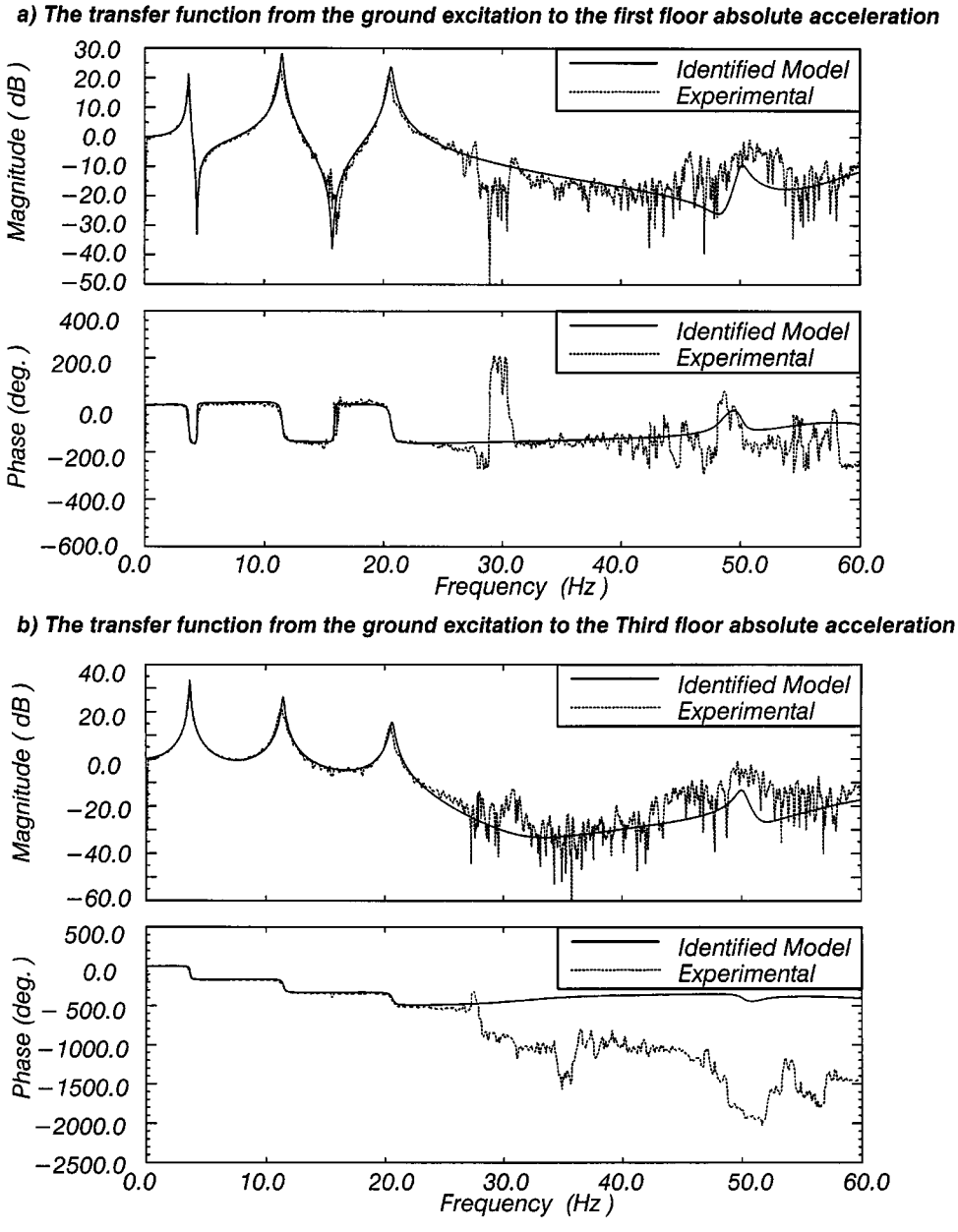


Figure 4. Sample comparison between the transfer functions from the ground excitation to (a) the first floor absolute acceleration and (b) the third floor absolute acceleration

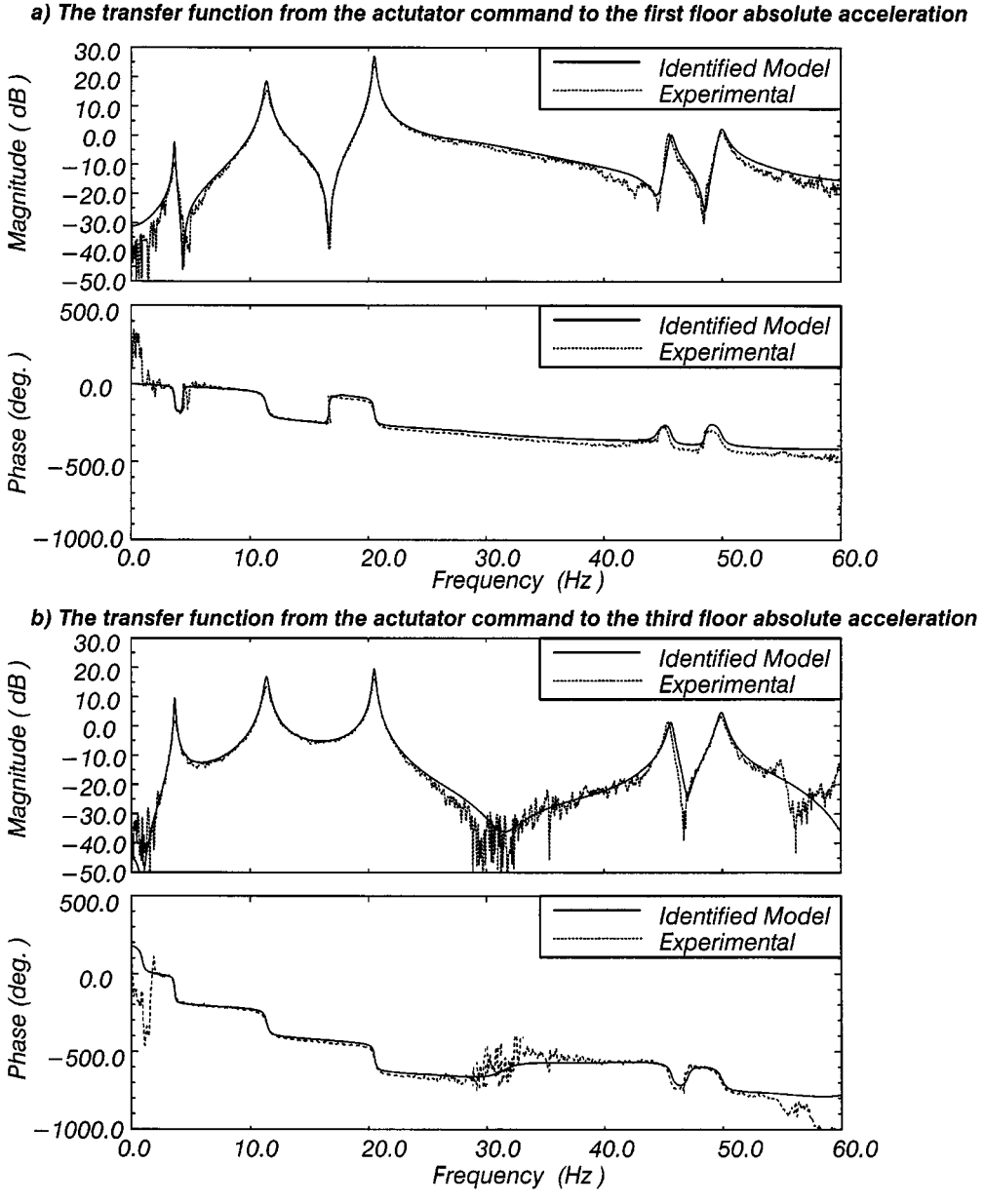


Figure 5. Sample comparison between the transfer functions from the actuator command to (a) the first floor absolute acceleration and (b) the third floor absolute acceleration

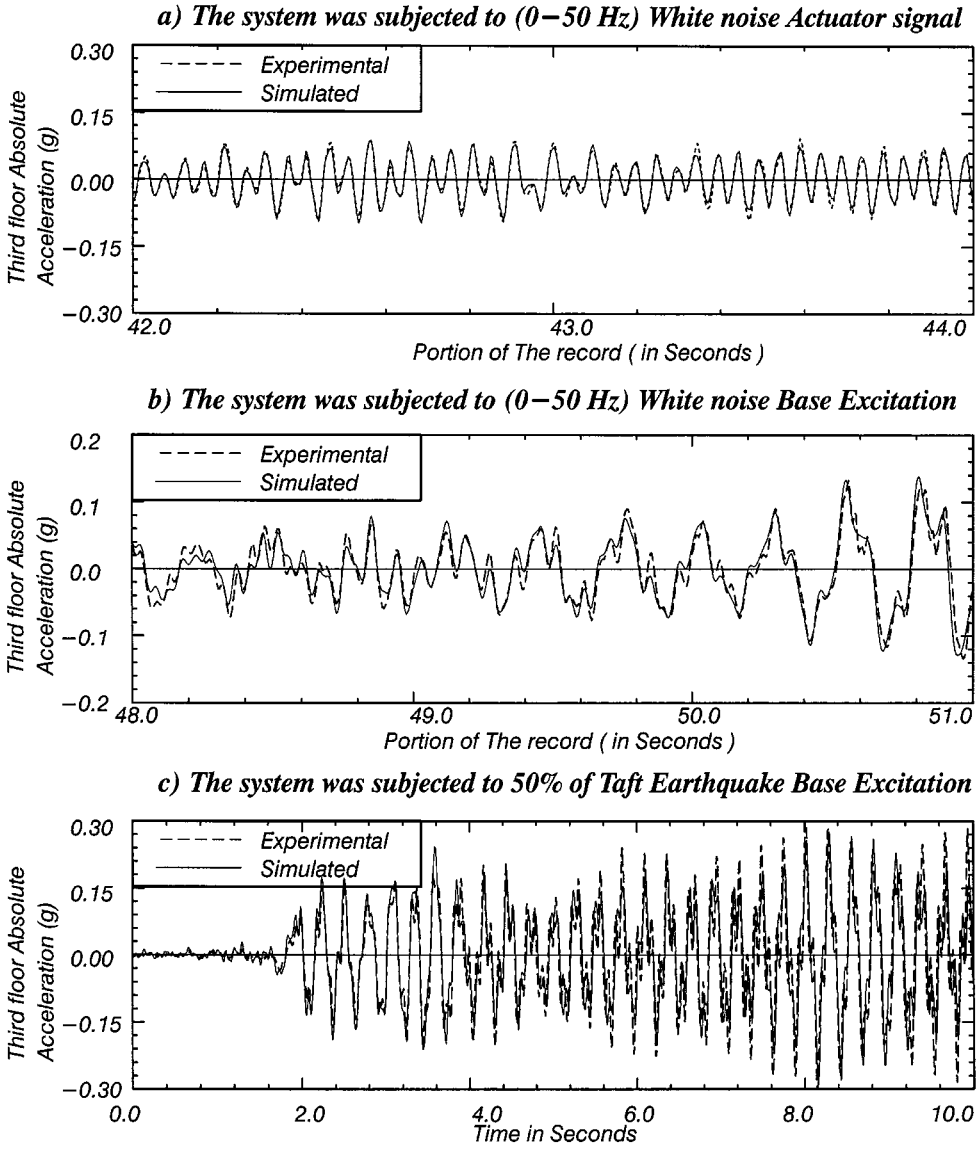


Figure 6. Comparison between the experimental and simulated results of the third floor absolute acceleration when the system was subjected to (a) (0–50 Hz) actuator commands, (b) (0–50 Hz) base excitation and (c) 50 per cent of the taft earthquake base excitation

NON-PARAMETRIC IDENTIFICATIONS: SYSTEM EMULATION USING NEURAL NETWORKS

Modelling the dynamical systems by using neural networks has been increasingly recognized as one of the system identification paradigms. The application of neural networks in system

identification is due to their generalization ability and their capability to describe the system model accurately, specially when they are trained from experimentally generated data containing the system model uncertainties, time delays, and the structure–environment interaction. The neural network modelling problem in system identification is to develop a neural network model that is capable of learning and predicting the functional mapping between the inputs and the outputs of an unknown linear or non-linear discrete-time multivariable dynamic system.

In this study, the experimental development of the neurocontrollers is associated with three independent neural network identifiers called emulator neural networks. These emulator neural networks are used on-line to train the neurocontroller from experimental results, a comprehensive description of the control method with the aid of the emulators is described in the companion paper of this study.[9] In addition to aiding in training the neurocontrollers, the emulator neural network corresponds to the identification of the system dynamics where it identifies the transfer functions of the structural model in a non-parametric manner (black-box). The emulator neural network is expected to learn the functional mapping between the inputs and outputs of a system. This can be described in the following function:

$$y_{k+1} = f(y_k, y_{k-1}, \ldots, y_{k-n}, u_{k-r+1}, u_{k-r}, u_{k-r-1}, \ldots, u_{k-r-m}) \tag{16}$$

where y_{k+1} represents the system response at time step $k+1$, u_k represents the present control command and r is a non-negative integer number controls the prediction capabilities of the emulator. Generally, the training process and the performance of the emulator degrades with larger values of r. The integers m and n are related to the degree of complexity and non-linearity of the underlying process represented by the function. A trial and error process usually produces reasonable values for n and m. It is important to note that the function in equation (16), which must be learned by the emulator neural network also includes the effects of the actuator dynamics, actuator saturation, time delays and the sampling period. It is, therefore, a nonlinear function, even if the structure itself behaves linearly.

The emulator neural network learns the relationships represented by equation (16). However, neural network representation is not exactly the same as the functions they learn. For this reason we use a different symbol to represent the trained neural networks.

$$y_{k+1} = \mathrm{NN}_e(y_k, y_{k-1}, \ldots, y_{k-n}, u_{k-r+1}, u_{k-r}, u_{k-r-1}, \ldots, u_{k-r-m}) \tag{17}$$

Whereas the mathematical functions are exact and universally true, the neural networks approximate these underlying functions over a limited range of interest. The uniqueness requirements for the neural networks are far more relaxed than for mathematical functions. For the neural network training to be successful, the underlying function must exist but need not be strictly unique. Moreover, even if the underlying function uniquely exists, the neural network architecture is not unique; more than one neural network can learn the same underlying function to within a given degree of accuracy over a limited range.

The neural network training method used in this study is an adaptive architecture determination method, which was originally developed in 1990[19,20] and, has since been modified and improved.[21] This method, combines the 'Quickprop' training algorithm proposed by Fahlman[22] and the dynamic node generation method proposed by Ash.[23] The essentials of the training method used in this study has been described in Joghataie *et al.*[21] The multilayer feedforward neural networks have been employed in this study. Experience has shown that the use of two hidden layers in the neural network model was sufficient for identification and control applications.

Two sets of three emulator neural networks have been developed and trained for each of the two neurocontroller designs (neurocontrollers U3A and UA presented in the companion paper). These emulator neural networks differ in their prediction capabilities, architecture and their sampling rates. In the first set, three emulators were used to train the first neurocontroller U3A. The first emulator of this set, referred to as EU3AI, was trained to learn a direct relationship between the system response and the present control command including the time delays, where the value of r in equation (17) is zero. With this emulator it was possible to predict the response from the past history of the response and the present control command and part of its past history. The second emulator, referred to as EU3AII, was trained to predict the system response from the past history of the system response and the past history of the control command, where $r = 1$. This emulator allowed the prediction of the system response one time step ahead. The third emulator, referred to as EU3AIII, had two time step prediction capabilities, where $r = 2$. The three emulators of the first set are represented using the neural network notation introduced by Ghaboussi *et al.*[24] The general form of this notation is:

$$\text{Output field} = \text{output field } \mathbf{NN}\ [\{\text{input field}\};\ \{\text{Neural Network architecture}\}] \tag{18}$$

Hence, the three emulators can be represented symbolically using this notation as follows:
Emulator EU3AI:

$$\ddot{\boldsymbol{x}}_k = \ddot{\boldsymbol{x}}_k^{\mathrm{P}}\,\mathbf{NN}\,[u_k,\ u_{k-1},\ u_{k-2},\ \ddot{\boldsymbol{x}}_{k-1},\ \ddot{\boldsymbol{x}}_{k-2},\ \ddot{\boldsymbol{x}}_{k-3};\ 12,\ 7,\ 7,\ 3] \tag{19}$$

Emulator EU3AII:

$$\ddot{\boldsymbol{x}}_k = \ddot{\boldsymbol{x}}_k^{\mathrm{P}}\,\mathbf{NN}\,[u_{k-1},\ u_{k-2},\ u_{k-3},\ \ddot{\boldsymbol{x}}_{k-1},\ \ddot{\boldsymbol{x}}_{k-2};\ \ddot{\boldsymbol{x}}_{k-3};\ 12,\ 9,\ 9,\ 3] \tag{20}$$

Emulator EU3AIII:

$$\ddot{\boldsymbol{x}}_k = \ddot{\boldsymbol{x}}_k^{\mathrm{P}}\,\mathbf{NN}\,[u_{k-2},\ u_{k-3},\ u_{k-4},\ \ddot{\boldsymbol{x}}_{k-1},\ \ddot{\boldsymbol{x}}_{k-2};\ \ddot{\boldsymbol{x}}_{k-3},\ 12,\ 11,\ 11,\ 3] \tag{21}$$

This representation describes the input and output field of the neural network model. The neural network symbol **NN** indicates that the output vector $\ddot{\boldsymbol{x}}_k^{\mathrm{P}}$ is predicted from neural network model. The information in the square brackets describes the neural network input and its architecture. The numbers following the semicolon describe the neural network architecture, i.e. number of layers and units in each layer.

In the second set, the three emulators were used to train the second neurocontroller UA. The first emulator of this set, is referred to as EUAI. The second emulator of this set is referred to as EUAII. The third emulator of this set is referred to as EUAIII. Similar to the first set of emulators, emulator EUAI learns the direct relationship between the control signal and the response. Whereas, emulator EUAII and EUAIII learn to predict the system response one time step ahead and two time steps ahead, respectively. These three emulators are symbolically represented by the following equations:
Emulator EUAI:

$$\ddot{x}_{3_k} = \ddot{x}_{3_k}^{\mathrm{P}}\,\mathbf{NN}\,[u_k,\ u_{k-1},\ u_{k-2},\ \ddot{x}_{3_{k-1}},\ \ddot{x}_{3_{k-2}},\ \ldots,\ddot{x}_{3_{k-6}};\ 9,\ 5,\ 5,\ 1] \tag{22}$$

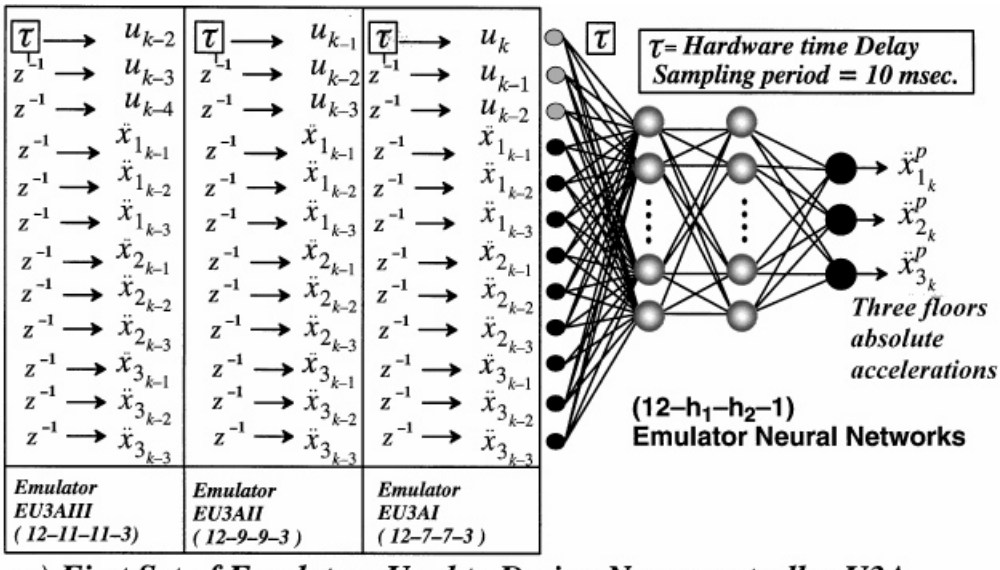
Emulator EUAII:

$$\ddot{x}_{3_k} = \ddot{x}_{3_k}^{\mathrm{P}}\,\mathbf{NN}\,[u_{k-1},\ u_{k-2},\ u_{k-3},\ \ddot{x}_{3_{k-1}},\ \ddot{x}_{3_{k-2}},\ \ldots,\ddot{x}_{3_{k-6}};\ 9,\ 6,\ 6,\ 1] \tag{23}$$

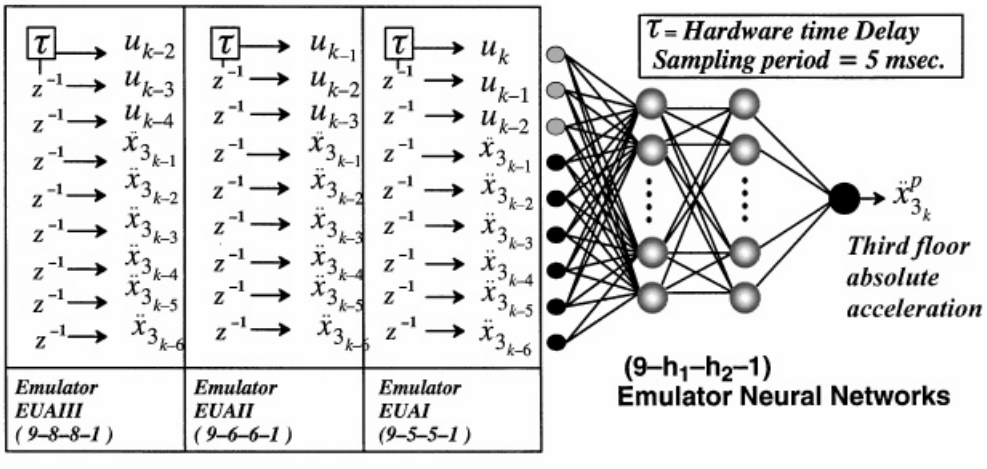
Emulator EUAIII:

$$\ddot{x}_{3_k} = \ddot{x}^{\mathrm{P}}_{3_k}\,\mathbf{NN}\,[u_{k-2},\, u_{k-3},\, u_{k-4},\, \ddot{x}_{3_{k-1}},\, \ddot{x}_{3_{k-2}},\, \ldots,\ddot{x}_{3_{k-6}};\, 9,\, 8,\, 8,\, 1] \tag{24}$$

The three emulators in each of the two sets were chosen to have the same m and n parameters given in equation (17). The first three emulators (EU3AI, EU3AII and EU3AIII) were developed to train the U3A neurocontroller. The architecture of these three emulators are shown in Figure 7, where the input layer has 12 nodes representing the values of the absolute accelerations



a) First Set of Emulators Used to Design Neurocontroller U3A



b) Second Set of Emulators Used to Design Neurocontroller UA

Figure 7. The architectures of the emulator neural networks used in this study

of the three floors and the control commands at the past three time steps. The control command setting in each emulator neural network reflects its prediction capabilities, and can be controlled by the values of r given in equation (17). Furthermore, the outputs of these emulators represent the absolute accelerations of the three floors at present time. The second set of these emulators (EUAI, EUAII and EUAIII) were developed to train neurocontroller UA. The architecture of these three emulators are shown in Figure 7. Their input layers have 9 nodes representing the absolute acceleration of the *third floor* at six past time steps and the control commands at three past time steps. The output of these emulators represents the third floor absolute acceleration at the present time. Again, the control command setting in each emulator neural network reflects its prediction capabilities and can be controlled by the values of r given in equation (17).

The number of the nodes in each hidden layer of the emulator neural networks were adaptively determined during the training method. The difference in the number of nodes in the hidden layers in the adaptive architecture determination reflects the degree of difficulty in learning the various prediction capabilities for the emulator neural networks.

Training of the six emulator neural networks has been performed on experimentally generated data from four earthquake simulator tests. The training data was collected when the system was set in motion with the following: (1) 25 s of 25 per cent of the amplitude of El Centro Earthquake record with time compressed factor of two; (2) 50 s of 0–50 Hz band limited white noise ground excitation; (3) 50 s of 0–50 Hz band limited white noise actuator command; and finally (4) 25 s of 25 per cent of the amplitude of El Centro earthquake record with time compressed factor of two and 25 s of 0–50 Hz band limited white noise actuator command. Neurocontroller U3A has a sampling period of 10 ms and neurocontroller UA has a sampling period of 5 ms. Consequently, the first set of the three emulators (EU3AI, EU3AII, and EU3AIII), associated with neurocontroller U3A, have a 10 ms sampling period therefore, 15 000 training cases were used in this set. The second set of the emulators (EUAI, EUAII and EUAIII) associated with neurocontroller UA have a 5 ms sampling period, and hence, 30 000 cases were used in their training.

Emulator neural networks evaluation

Evaluation of the prediction capabilities of the trained emulator neural networks have been presented in frequency and time domain. In the frequency domain, the effectiveness of the emulators in reproducing and predicting the system transfer functions have been investigated and studied. The six emulators (EU3AI, EUAI, EU3AII, EUAII, EU3AIII and EUAIII) have been employed to reproduce the transfer functions from experimentally generated data, where the inputs to each emulator have been collected directly from the instrumented sensors on the specimen. Figure 8 shows the procedure to produce the system transfer functions using the three emulator neural networks in parallel-series fashion with a control signal of 0–50 Hz band limited white noise.

Clearly, the sample illustration in Figures 11 and 14 show that the emulators are able to predict the transfer functions very well. However, as mentioned earlier, the performance of the emulators are slightly degraded with the increase of the values of r in equation (17). This is likely due to two reasons: (1) as r increases, the uncertainty in predicting further into the future increases and (2) the input information in the second and the third emulators contain the output of the first and the second emulators which already have some small prediction error, therefore, the predicted response compounds the accumulated error.

In time domain, the performance of the emulator neural networks were evaluated by comparing their response with the experimental results. This evaluation was performed for three different

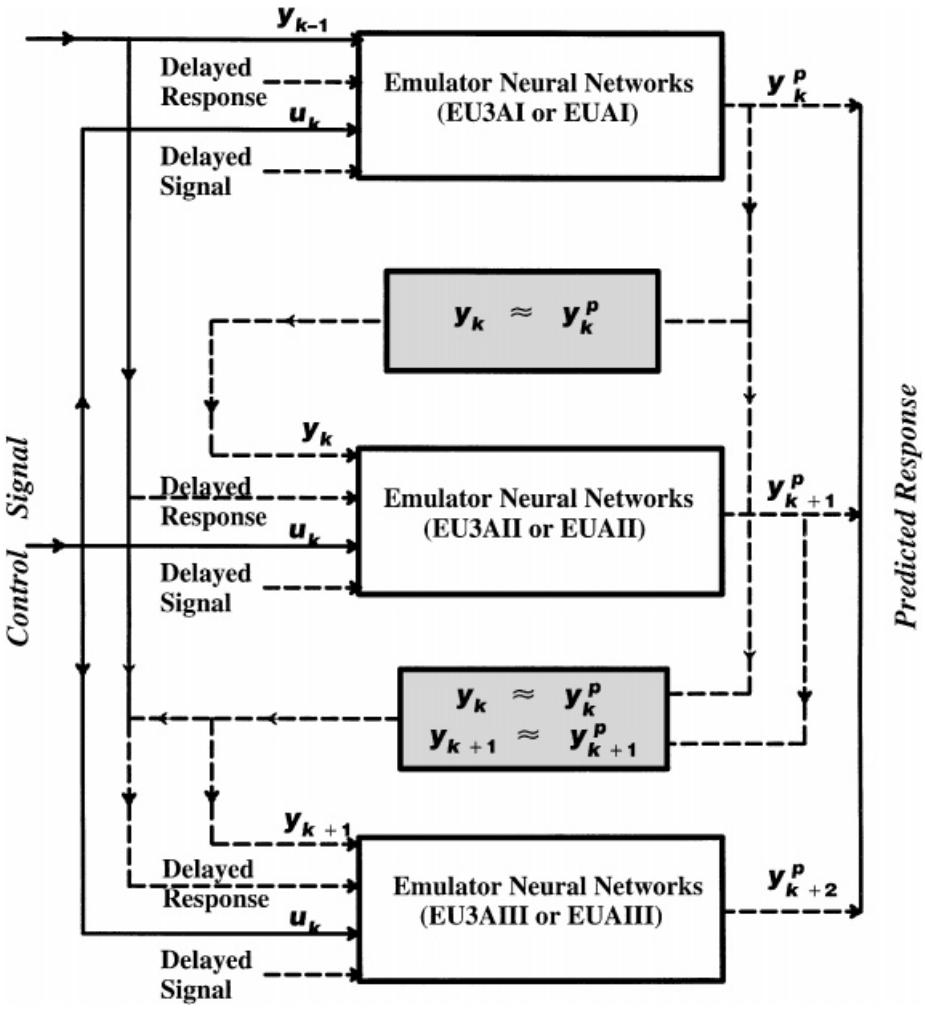


Figure 8. The emulators neural networks in parallel–series fashion

cases: (1) 50 per cent of 1952 Taft earthquake record with zero control command for the six emulators; (2) 20 per cent of 1994 Northridge earthquake record at Santa Monica—City Hall, with zero control command for the first set of emulators (EU3AI, EU3AII and EU3AIII); and (3) 50 s period 0–50 Hz of band limited white noise actuator command for the second set of emulators (EUAI, EUAII and EUAIII). The results for these experiments are shown in Figures 9, 10, 12 and 13. The same phenomenon of performance degredations with the increase in the value of the r is noticed, for the same reasons as stated before. Clearly, the emulator neural networks have been able to learn the transfer functions from the control command to the system response very well, and to reproduce the structural response under different seismic excitations very accurately. In the evaluation of the emulator neural networks it has been intended to use validation experimental data different than the experimental training data. Clearly, as evidenced from the

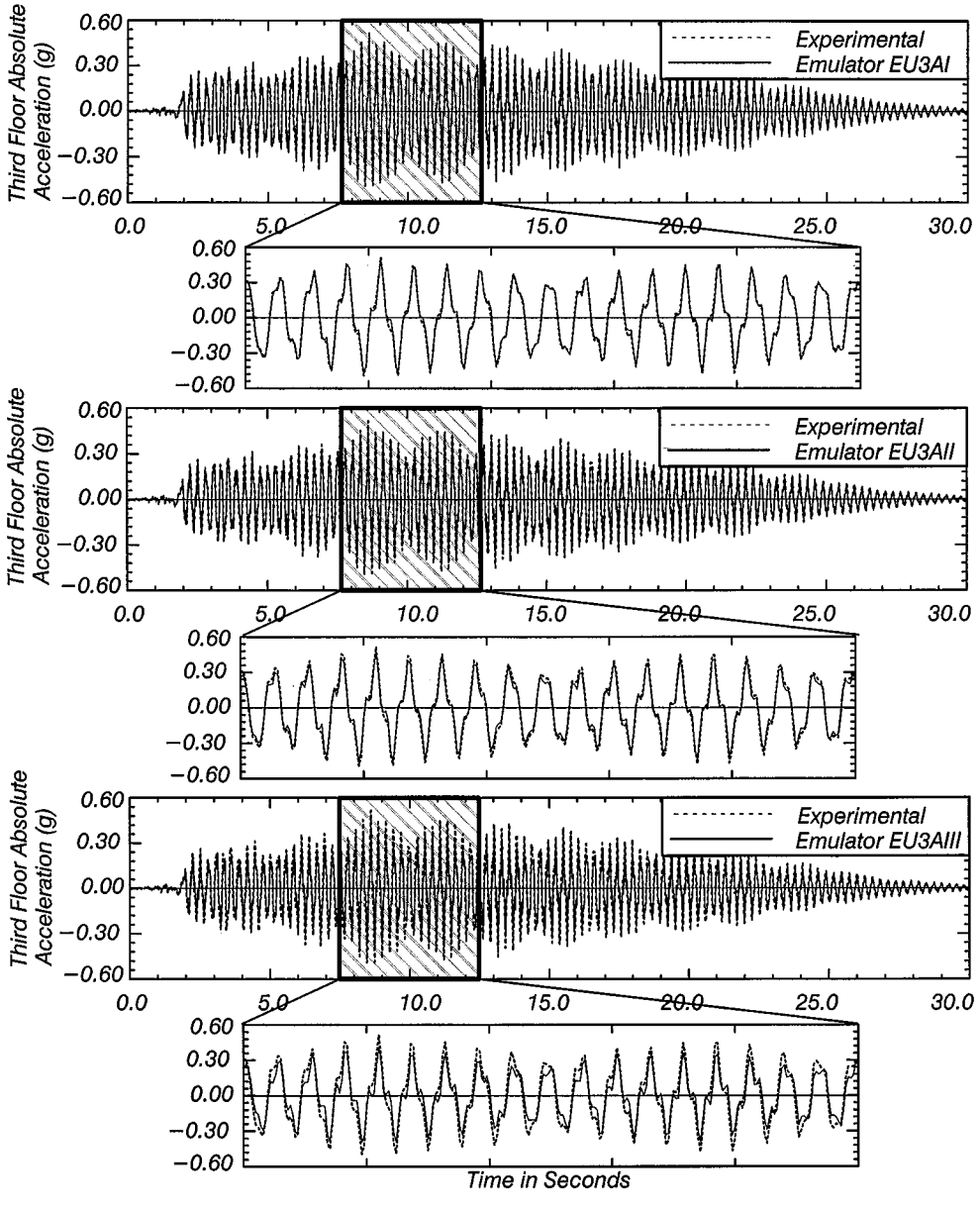


Figure 9. Comparison of the experimental and emulators response (using EU3AI, EU3AII and EU3AIII). The structure has been excited by 50 per cent of Taft earthquake record

experimental results, the emulators were capable of predicting the response for novel cases that were not included in the training cases of the neural networks. This is always true when the neural networks are used, which makes the emulator neural networks independent of the training cases and generalized model for the system under study.

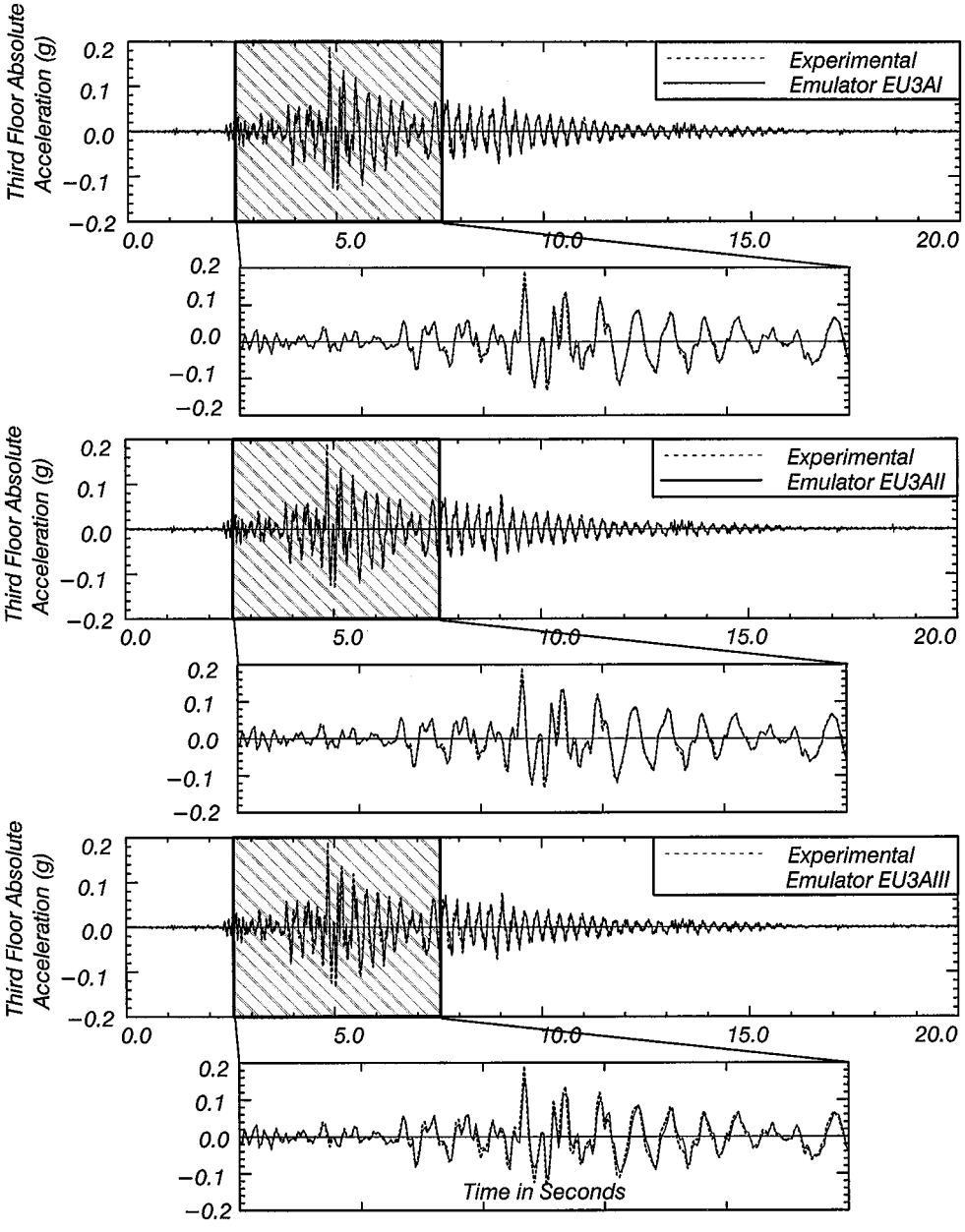


Figure 10. Comparison of the experimental and emulators response (using EU3AI, EU3AII and EU3AIII). The structure has been excited by 20 per cent of Northridge record

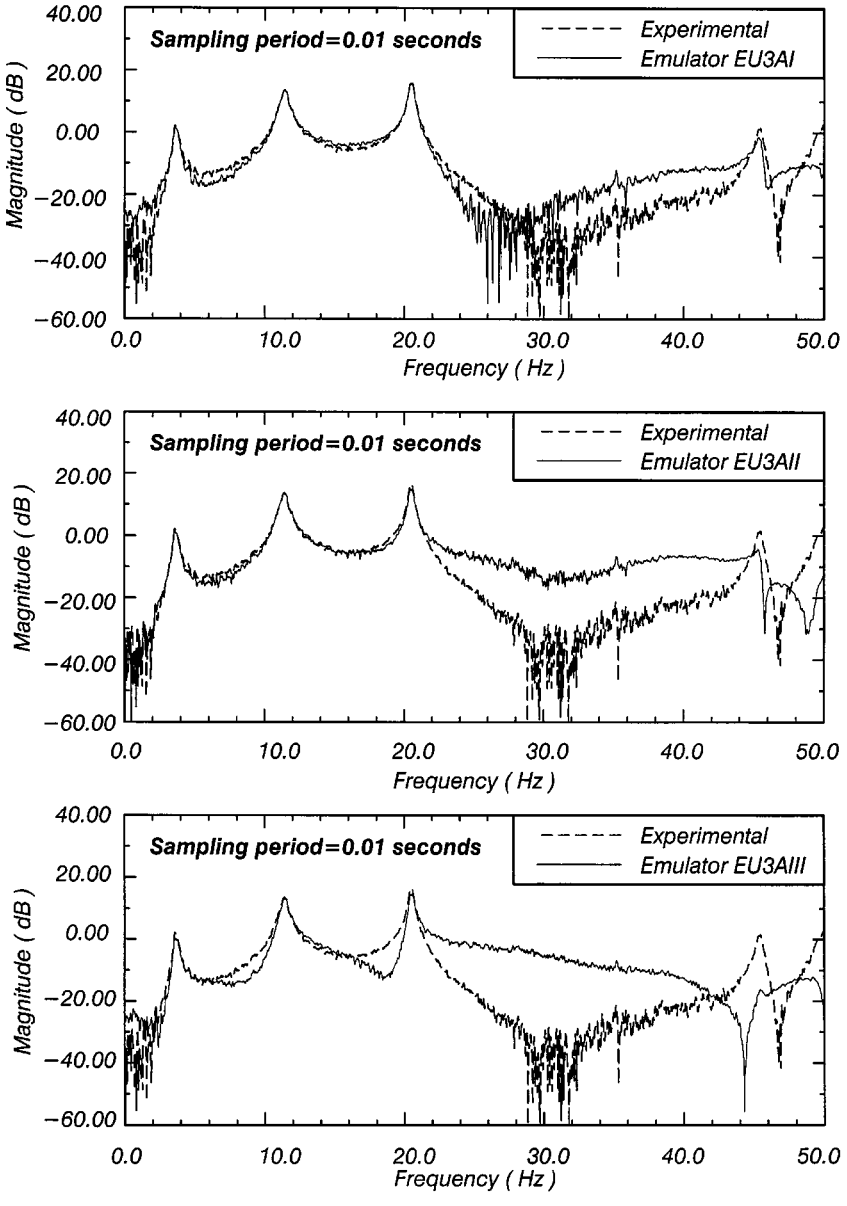


Figure 11. Comparison of the experimental transfer function of the actuator command to the *third floor* absolute accelerations using the first set of the three emulator neural networks (EU3AI, EU3AII and EU3AIII)

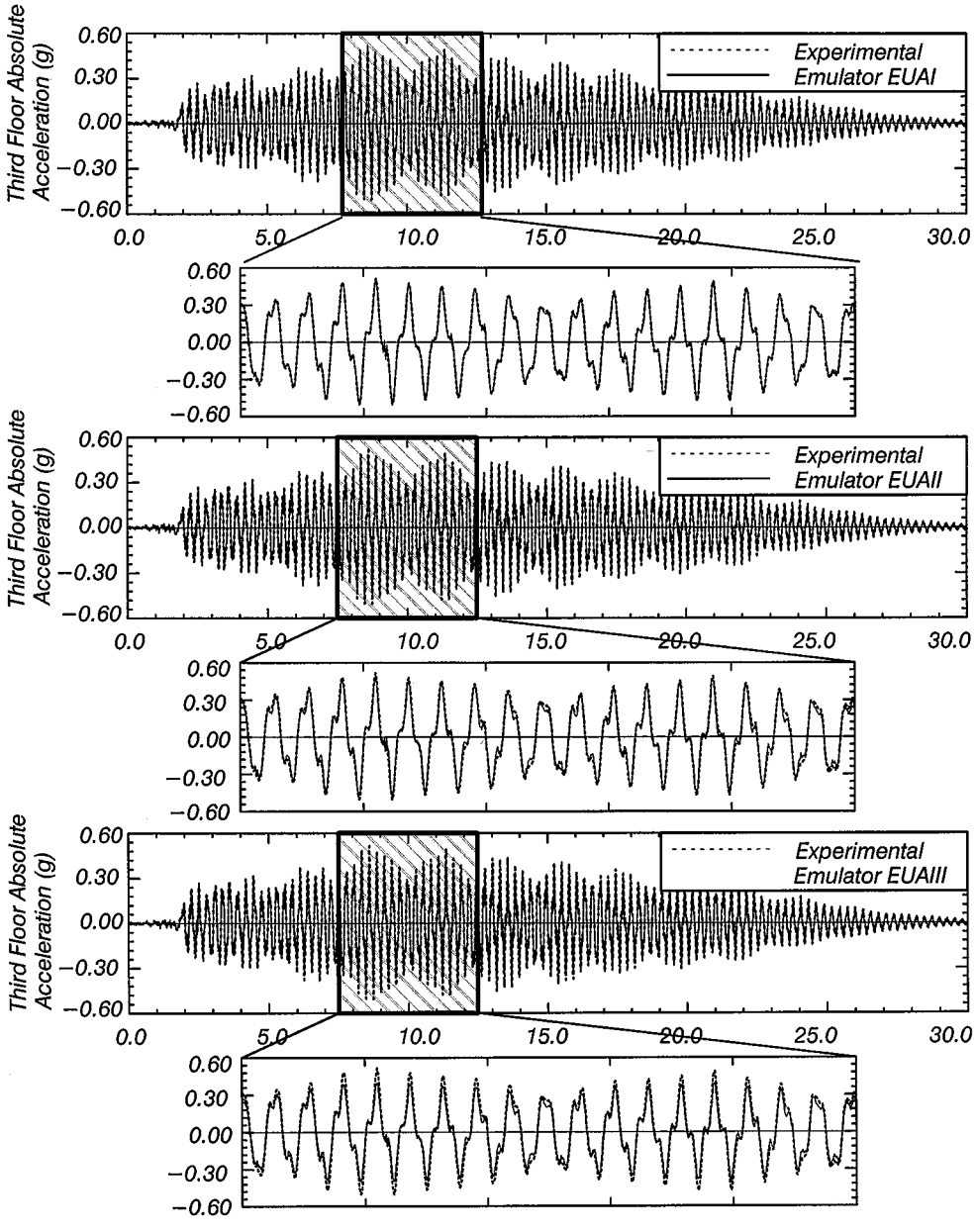


Figure 12. Comparison of the experimental and emulators response (using EUAI, EUAII and EUAIII). The structure has been excited by 50 per cent of Taft earthquake record

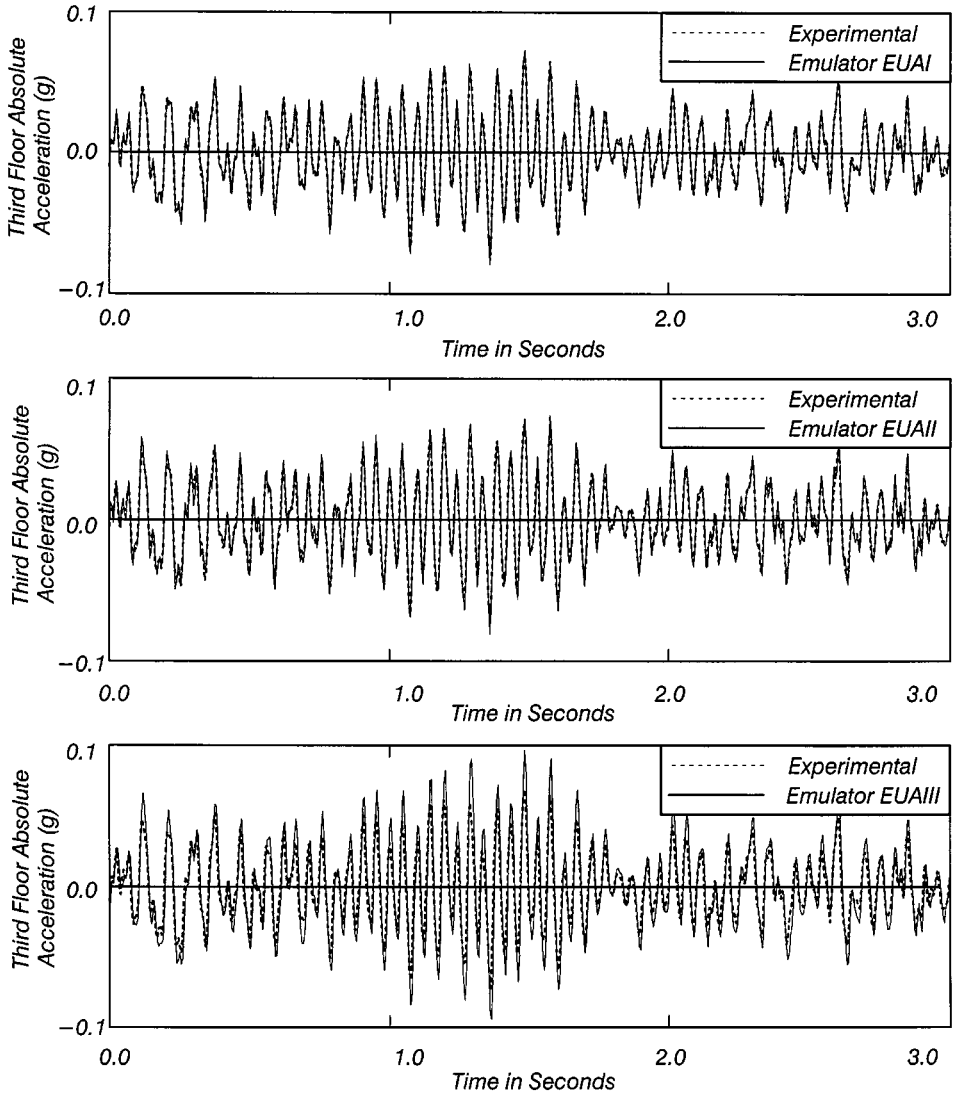


Figure 13. Comparison of the experimental absolute acceleration of the *third floor* and the reproduced response using the second set of the three emulator neural networks (EUAI, EUAII and EUAIII). The structure was excited by white noise actuator commands

CONCLUDING REMARKS

Experimental verifications of a neural networks based system identification has been presented and evaluated. The neural network models used for the system identification were called emulator neural networks. These emulators were developed for use in an experimental study of a recently

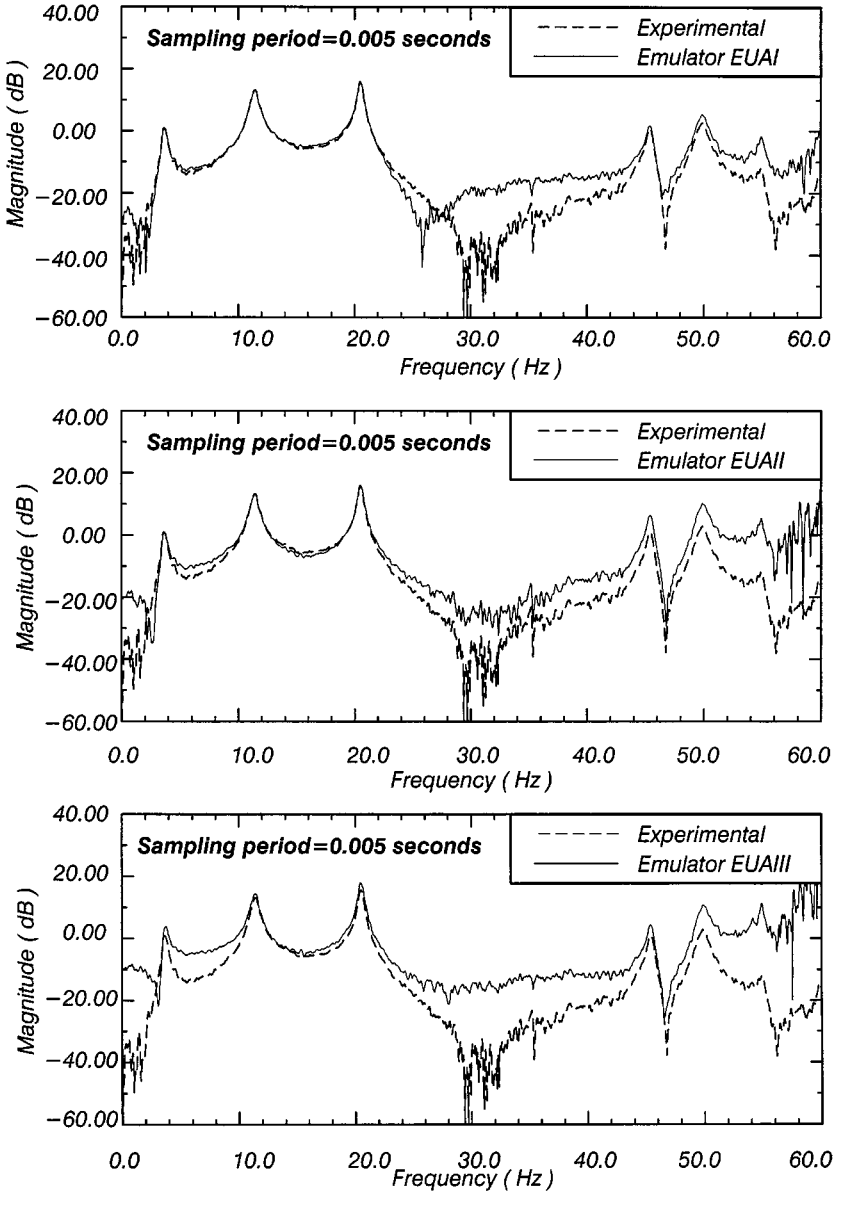


Figure 14. Comparison of the experimental transfer function of the actuator command to the *third floor* absolute accelerations using the second set of the three emulator neural networks (EUAI, EUAII and EUAIII)

developed structural control method using neural networks. Six different emulators were trained and verified. These emulators have different architectures, prediction capabilities and sampling rates. The experiments were performed on the earthquake simulator at the University of Illinois at Urbana—Champaign. First, the system was identified in the time domain and the estimated parameters were used in the frequency domain methods in a parametric identification method. In

a second method, the system was modelled and identified using multiple emulator neural networks, intended for use in the neurocontrol design. The experimental set-up was fully described and presented. The experimental validation of the mathematical model has been established in the time and frequency domains. The multiple emulator neural networks performance was demonstrated experimentally and shown to be independent of the training data.

ACKNOWLEDGEMENTS

The research reported in this paper was funded by National Science Foundation Grant CMS-95-003209. This support is gratefully acknowledged.

REFERENCES

1. L. L. Chung, A. M. Reinhorn and T. T. Soong, 'Experiments on active control of seismic structures', *J. Engng. Mech. Div.* ASCE **114**, 241–256 (1988).
2. J. Rodellar, L. L. Chung, T. T. Soong and A. M. Reinhorn, 'Experimental digital control of structures', *J. Engng. Mech. Div.* ASCE **115,** 1245–1261 (1989).
3. T. T. Soong, A. M. Reinhorn and Y. P. Wang, 'Full scale implementation of active control, design and simulation', *J. Struct. Engng. Div.* ASCE **117**(11), 3516–3536 (1991).
4. S. J. Dyke, B. F. Spencer, B. Quast and M. K. Sain, 'Acceleration feedback control of MDOF structures', *J. Engng. Mech. Div.* ASCE **122**(9), 907–918 (1996).
5. S. F. Masri, A. G. Chassiakos and T. K. Caughey, 'Structure—unknown non-linear dynamic systems: identification through neural networks', *J. Smart Mater. Struct.* **1**, 45–56 (1992).
6. J. P. Conte A. J. Durrani and R. O. Shelton, 'Response emulation of multi-storey buildings using neural networks', *Proc. 1st World Conf. on Structural Control*, Los Angeles, CA 1984, pp. wp1-59–wp1-68.
7. K. Bani-Hani, J. Ghaboussi and S. P. Schneider, 'Experimental study of neural networks based-structural control', *Proc. 6th U.S. National Conf. on Earthquake Engineering*, Seattle, Washington 1998.
8. K. Bani-Hani, J. Ghaboussi and S. P. Schneider, 'Experimental study of structural control using neural networks', *Proc. 2nd World Conf. on Structural Control*, Kyoto, Japan, 1998.
9. K. Bani-Hani, J. Ghaboussi and S. P. Schneider, 'Experimental study of identification and control of structures using neural network: Part 2: control', *Earthquake Engng. Struct. Dyn.*, **28**, 1019–1039 (1999).
10. M. A. Sozen, S. Otani, P. Gulkan and N. N. Nielsen, 'The university of Illinois earthquake simulator', *Proc. 4th World Conf. on Earthquake Engineering*, Santiago, Vol. III, Session B-5, Chile, January 13–18, 1969, pp. 139–150.
11. R. S. Miller, H. Krawinkler and J. M. Gare, 'Model tests on earthquake simulators development and implementation of experimental procedure', *Report No. 39*, *Dept. of Civil Eng.*, Stanford University, CA, 1979.
12. R. W. Clough and D. T. Tang, 'Earthquake simulator study of a steel frame structure, Vol. 1: experimental results', *EERC Report No. 75-6*, University of California, Berkeley, CA, 1975.
13. T. T. Soong, A. M. Reinhorn and J. N. Yang, 'A standardized model for structural control experiments and some experimental results', in H. H. E. Leipholz (ed.), *Proc. 2nd Int. Symp. on Structural Control*, University of Waterloo, Ont., Canada, 15–17 July 1985, Martinus Nijhoff, Amsterdam, 1987.
14. L. Ljung, *System Identification*: *Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
15. W. D. T. Davis, *System Identification for Self Adaptive Control*, Wiley, New York, 1970.
16. J. Schoukens and R. Pintelon, *Identification of Linear Systems*: *A practical Guideline to Accurate Modeling*, Pergamon Press, New York, 1991.
17. K. G. McConnell, *Vibration Testing Theory and Practice*, Wiley, New York, 1995.
18. The Math Works, Inc., *MATLAB*, Natick, MA, 1994.
19. J. Ghaboussi, J. H. Garrett and X. Wu, 'Material modeling with neural networks', *Proc. Int. Conf. on Numerical Methods in Engineering*: *Theory and Applications*, Swansea, U.K., 1990, pp. 701–717.
20. J. Ghaboussi, J. H. Garrett and X. Wu, 'Knowledge-based modeling behavior with neural networks', *J. Engng. Mech. Div.* ASCE **117**(1), 132–153 (1991).
21. A. Joghataie, J. Ghaboussi and X. Wu, 'Learning and architecture determination through automatic node generation', *Proc. Int. Conf. on Artificial Neural Networks in Engng.*, St. Louis, 1995.
22. S. E. Fahlman, 'Faster learning variations on error back propagation: an emperical study', *Carnegie-Mellon Summer Workshop on Neural Networks*, 1988.
23. T. Ash, 'Dynamics node creation in backpropagation networks', *Proc. Int. Joint Conf. on Neural Network* (IJCNN), Vol. VII, Washington D.C. June, 1989, pp. 623–629.
24. J. Ghaboussi, D. A. Pecknold, M. Zhang and R. M. HajAli, 'Autoprogressive training of neural network constitutive models', *Int. J. Numer. Meth. Engng.*, **42**, 105–126 (1998).

